

Exam 1

February 9, 2016

Create a markdown file in RStudio, and call it `YourLastName_Exam1`. Answer all of the exam questions in this file, with your r code for each problem inserted in a separate chunk. When you are finished, copy your file to a flash drive and bring it to the front to turn it in. Be sure it compiles properly before you turn it in, and remember to turn it in before you leave.

1. Write a short chunk of r code that clears all variables from memory and sets the random number seed to a value of 1000. **(4 points)**

```
rm(list=ls())
set.seed(1000)
```

2. In population genetics, the Hardy Weinberg equation gives the expected genotype frequencies in a randomly mating population, given an initial frequency p for an allele A , and frequency q for the alternate allele B . The essential equations are:

$$p + q = 1$$

$$f(AA) = p^2$$

$$f(AB) = 2pq$$

$$f(BB) = q^2$$

- a) Typeset these equations in your markdown file so they look the same as they do here. **(4 points)**

\$\$ p + q = 1 \$\$

\$\$ f(AA) = p^2 \$\$

\$\$ f(AB) = 2pq \$\$

\$\$ f(BB) = q^2 \$\$

- b) Write a function `HardyWeinberg` that takes as input the value of p , and returns a vector containing $(f(AA), f(AB), f(BB))$. In your function, use a random uniform number to set an initial value for p . **(8 points)**

```
HardyWeinberg <- function(p=runif(1)){
  q <- 1 - p
  v <-vector(mode="numeric",3)
  names(v) <- c("f(AA)", "f(AB)", "f(BB)")
  v[1] <- p^2
  v[2] <- 2*p*q
  v[3] <- q^2
  return(v)
}
```

c) Call your function twice, once using the default values, and once using $p = 0.1$. (6 points)

```
HardyWeinberg()
```

```
##      f(AA)      f(AB)      f(BB)
## 0.1075044 0.4407485 0.4517470
```

```
HardyWeinberg(p=0.1)
```

```
## f(AA) f(AB) f(BB)
##  0.01  0.18  0.81
```

d) Call your function with $p = 0.5$, and save your results to a variable z . Print the answer to the screen, so that it says “The allele frequencies are: *<result of your function>*” and then prints a carriage return. (4 points)

```
z <- HardyWeinberg(p=0.5)
cat("The allele frequencies are: ", z, "\n")
```

```
## The allele frequencies are:  0.25 0.5 0.25
```

3. a) Using `runif`, create an initial vector of 1000 random uniform values that range from -2 to +2 (use R's help system if you need help doing this). **(4 points)**

```
z <- runif(1000, min=-2,max=2)
```

- b) Create a second vector (using the `vector` function) to hold 1000 character strings. Using a `for` loop, fill this vector with character strings. Whenever the value in your initial vector is < 0 , assign to the second vector the character string "negative". Whenever the value in your initial vector is > 0 , assign to the second vector the character string "positive". **(4 points)**

```
z2 <- vector(mode="character",1000)
for (i in 1:1000) {
  if(z[i] >0) z2[i] <- "positive" else z2[i] <- "negative"
}
```

- c) print out the last six elements of your two vectors. **(4 points)**

```
print(z[994:1000])
```

```
## [1]  1.00877982  0.09952362 -0.60774764  1.77639659 -1.09890711  0.57554240
## [7]  1.02176246
```

```
print(z2[994:1000])
```

```
## [1] "positive" "positive" "negative" "positive" "negative" "positive"
## [7] "positive"
```

- d) Recreate the string vector, this time using a single call to the function `ifelse`. ****(4 points)****

```
z3 <- ifelse(z>0, "positive", "negative")
print(z[994:1000])
```

```
## [1]  1.00877982  0.09952362 -0.60774764  1.77639659 -1.09890711  0.57554240
## [7]  1.02176246
```

```
print(z3[994:1000])
```

```
## [1] "positive" "positive" "negative" "positive" "negative" "positive"
## [7] "positive"
```

4. a) Pick any R function that you have not used in any of the previous problems. Write a user-friendly paragraph explaining how the function is used, and illustrate it with some code. Be sure to illustrate all of the options that are available for this function. If you have already written something for your own "help system", you can cut and paste that and indicate it is your own writing. **(6 points)**
- b) Go out onto the internet and find a blog or internet entry that uses or illustrates your function. Cut and paste the example into an R chunk. In your Rmarkdown, provide a link to the website where you took this information from. **(6 points)**